REMARKS

Claims 1-4, 20-22 and 39-43 are pending in the application. Claims 5-19 and 23-38 are currently withdrawn. Claims 1, 3, 20 and 21 are amended herein. Claims 39-43 are new. Claims 1-4 and 20-22 currently stand rejected under 35 U.S.C. §102(e) based on Brendel, US Pub. No. 2005/0010754 (hereinafter "Brendel").

As a threshold matter, the finality of the Office Action is clearly improper under MPEP §706.07(c), and should therefore be withdrawn.

More specifically, MPEP §706.07(b) states "it would not be proper to make final a first Office action in a continuing or substitute application where that application contains new material which was presented in the earlier application after final rejection or closing of prosecution but was denied entry because (A) new issues were raised that required further consideration and/or search or (B) the issue of new matter was raised." Here, the RCE filed Feb. 22, 2007 included subject matter (claim amendments) that was refused entry in the Advisory Action, mailed Feb. 8, 2007, on the grounds these amendments "raise[d] new issues that would require further consideration and/or search." Therefore, MPEP §706.07(b) applies, and the finality of the Office Action should be withdrawn.

Turning to the merits, independent Claims 1, 3, 20, and 21 have been amended to recite one or more patentable distinctions over Brendel. New Claims 39-43 recite or incorporate similar patentable distinctions over Brendel. As a result of these amendments, each of Claims 1-4, 20-22 and 39-43 patentably distinguishes over Brendel, and should be allowed.

Consider Claim 1. As amended, that claim calls for "first logic in the network for determining if any of a **plurality of persistence policies**, comprising at least one first persistence policy and at least one second persistence policy, is applicable to a service request, and, if so, allocating the resource to the request based on application of the persistence policy determined to be applicable." (Emphasis added).

Through application of a plurality of persistence policies, the claimed system is able to handle a **plurality of different types of service requests**. This advantage is expressed in the recitation in the claim calling for this first logic to be configured so that:

> "when a first type of service request is received, the first logic determines if the at least one first persistence policy is applicable;

> when a second type of service request is received, the first logic determines if the at least one second persistence policy is applicable;
>
> when the first type of service request is received but it is determined that the first persistence policy is inapplicable, the first logic determines if the at least one second persistence policy is applicable"

Only if none of the plurality of persistence policies is applicable does the system consider applying a load balancing policy. This is expressed in the recitation calling for "second logic in the network for allocating the resource to the request based on application of a load balancing policy only if none of the plurality of persistence policy is determined to be applicable as determined by the first logic."

The claimed system also requires **sharing of persistence policies between different types of resource requests**. This is expressed in the following recitation calling for the first logic to be configured so that:

> "when a second type of service request is received, the first logic determines if the at least one second persistence policy is applicable;
>
> when the first type of service request is received but it is determined that the first persistence policy is inapplicable, the first logic determines if the at least one second persistence policy is applicable"

This sharing of persistence policies amongst different types of resource requests gives rise to efficiencies because it allows the two different types of resource requests to share, at least in part, the same mechanism for allocating resources based on persistence.

The support for these amendments (and new Claims 39-43) is provided, for example, by pages 16-17, 19-21 of the application, describing an embodiment where the system "supports both L4 and L5-7 transactions through the same mechanism." *see* page 19, line 30, to page 20, line 1, which refers to the ability of the system to handle both OSI layer 4 service requests and OSI layer 5-7 service requests through the same allocation mechanism.

On page 19, line 7, the specification refers to an L4 service request as "non content aware" because it is not aware of content, *i.e.*, L5-7, and refers to a L5-7 service request as "content aware" because it is aware of content, *i.e.*, L5-7. Persistence policies that are aware of content, such as cookie- or session-based persistence policies, *see* page 20, line 19, may also be referred to as "content aware," while persistence-policies that are not aware of content, such as client-based persistence policies, *see* page 20, lines 19-20, and page 31, lines 6-7, may be referred to as "non-content-aware."

For a L4 service request, the system checks for persistence using a client-based persistence policy. *See* page 21, lines 11-12 ("Stickiness for an L4 request is based on the client identity."). For a L5-7 service request, the system first checks for persistence using a cookie-based or session-based persistence policy. *See* page 21, lines 12-13 ("For an L5-7 request, cookie-based or session-based stickiness is attempted.). If that is unsuccessful, the system then checks the L5-7 service request for persistence using a client-based persistence policy, the same persistence policy used to check for persistence of the L4 service request. *See* page 28, lines 15-17 ("If the access fails to yield an entry corresponding to the first index, a second index is derived from the client IP address of the corresponding packet, and used to access the history table. If this access yields an entry corresponding to the entry, the corresponding server is allocated to the request."). The cookie-based and session-based persistence policies are not attempted for a L4 service request since these policies are based on L5-7 packet content, which is lacking in a L4 service request. *See* page 19, page 7, referring to a L4 service request as "non-content aware."

Examples of cookie-based persistence policies include self-identification stickiness, cookie hashing stickiness and cookie-ID based persistence. See page 16, lines 13-14. An example of a session-based persistence policy is session-ID based persistence. *See* page 16, lines 14-15. An example of a client-based persistence policy is client IP-address based persistence. *See* page 17, lines 1-3.

Only if none of the plurality of persistence policies is applicable to the request does the system utilize a load-balancing policy to allocate the resource. *See* page 21, lines 9-10.

Based on the foregoing, it can be seen that, in this embodiment, a client-based persistence policy is used to check for persistence both when a L4 service request is received, and also when a L5-7 service request is received, provided it has been determined that a cookie-based or session-based persistence policy is inapplicable to the L5-7 request. By sharing the same persistence policy between the different types of service requests, the **same persistence allocation mechanism** can be shared, at least in part, between the L4 and L5-7 service requests.

See page 19, line 30, to page 20, line 1 ("The policy engine 40 supports both L4 and L5-7 transactions through the same mechanism").

In contrast to the foregoing, in Brendel, an incoming request is parsed to determine if it represents an encrypted or a clear-text connection. *See* par. 68. If the request represents a clear-text connection, the request is further parsed for a cookie. *See* par. 69. If a cookie is present, and a server is identified by the cookie value or encoded into the cookie, the server is assigned to the request. *Id.* Otherwise, a server is assigned to the request through random or default load-balancing. *See* par. 70.

If the request represents an encrypted connection, any cookie in the unencrypted request is unavailable. *See* par. 71. Therefore, the server extracts a SSL session ID from the request and compares it with entries in a table. *See* par. 72. If a match is found, the server identified by the matching entry is assigned to the request. *Id.* If no match is found, a server is assigned to the request through random or default load-balancing. *See* par. 73.

Significantly, in Brendel, there is **no** sharing of persistence policies between different types of service requests. In the case of a request representing a clear-text connection, Brendel checks for persistence using a cookie-based policy, and, if that is unsuccessful, it assigns a server to the request based on random or default load-balancing, without applying another persistence policy. Similarly, in the case of a request representing an encrypted connection, Brendel checks for persistence using a session-ID based policy, and, if that is unsuccessful, it assigns a server to the request based on random or default load-balancing, again without applying another persistence policy. Therefore, in Brendel, the same persistence allocation mechanism cannot be shared at all between the requests representing clear-text connections and those representing encrypted connections. Instead, two distinct persistence allocation mechanisms must be used, one for cookie-based persistence, and the other for session-ID based persistence.

Moreover, nothing in Brendel suggests sharing the same persistence policies between different types of service requests. Therefore, Claim 1 is patentable over Brendel because it requires sharing of persistence policies between different types of service requests. All the other claims recite or incorporate this same or a similar requirement. Consequently, all claims are patentable over Brendel.

For all the foregoing reasons, reconsideration of and withdrawal of all outstanding rejections is respectfully requested. The Examiner is earnestly solicited to allow all claims, and pass this application to issuance.

To expedite allowance of this case, the Examiner is earnestly invited to call Robert C. Laurenson at (949) 759-5269.

Respectfully submitted,

Date: June 8, 2007

____/Robert C. Laurenson/__________
Robert C. Laurenson (Reg. No. 34,206)

HOWREY LLP
2941 Fairview Park Drive, **Box No. 7**
Falls Church, VA 22042
FAX No. (703) 336-6950
Telephone No. (949) 759-5269